Two-Level Scheduling Algorithm for Different Classes of Traffic in WiMAX Networks

Zeeshan Ahmed
LUNAM Université, IRCCyN,
CNRS UMR 6597, Polytech′ Nantes,
rue Christian Pauc - BP 50609 -
44306 Nantes Cedex 3 FRANCE
Email: zeeshan.ahmed@univ-nantes.fr

Salima Hamma
LUNAM Université, IRCCyN,
CNRS UMR 6597, Polytech′ Nantes,
rue Christian Pauc - BP 50609 -
44306 Nantes Cedex 3 FRANCE
Email: salima.hamma@univ-nantes.fr

***Abstract*—The IEEE 802.16 standard is one of the most promising broadband wireless access systems. The standard incorporates a QoS architecture that supports both realtime and non-realtime applications. To provide QoS three data schedulers are furnished by the architecture. However, the working of the schedulers are not defined by the standard. Some researchers have attempted to fill this gap by providing different scheduling schemes. However, no scheme has yet been adapted by the standard and the area is still open for new research. In this article we propose Two-Level Scheduling Algorithm (TLSA) that ensures QoS for all service classes, while avoiding starvation of lower priority classes. Furthermore, it ensures fair resource allocation among flows of the same class. The simulation results show that the algorithm is effective and efficient.**

I. INTRODUCTION

The IEEE 802.16 standard [1], also known as Worldwide Interoperability for Microwave Access (WiMAX), is a broadband wireless access (BWA) technology that supports mobility and high speed data transfer. WiMAX is a candidate to become the standard for the fourth generation of digital cellular networks.

WiMAX provides an extensive support for providing QoS to both multimedia and data applications. To support different types of applications, five service classes are provided by the standard. (i)*Unsolicited Grant Service(UGS):* specifically designed for constant bit-rate services, such as T1/E1 emulation and VoIP without silence suppression (ii)*Extended Realtime Polling Service(ertPS):* built on the efficiency of both UGS and rtPS. Suitable for applications such as VoIP with silence suppression. (iii) *Realtime Polling Service(rtPS):* designed for realtime services that generate variable size data packets on periodic basis, such as MPEG video (iv) *Non Realtime Polling Service(nrtPS):* designed for delay tolerant services that generate variable size data packets on regular basis (v) *Best Effort(BE) Service:* designed for applications without any QoS requirements such as HTTP service.

An essential functionality of a QoS architecture is the scheduling of network traffic. The standard provides three schedulers: (i) Base station (BS) uplink scheduler (ii) BS downlink scheduler (iii) and Subscriber station (SS) scheduler. The functions of these schedulers are defined but their working is not defined by the standard. Therefore vendors and service providers can choose scheduling mechanisms that best suit their needs.

The most challenging part of scheduling is done by BS uplink scheduler. The role of uplink scheduler is to grant uplink access to SS according to their QoS requirements. However the scheduler does not have access to input data queues, which are maintained at the SS. The scheduler cannot know the size of individual packets that are stored in those queues and therefore scheduling has to be done according to some estimates.

In this paper, we propose an efficient algorithm for uplink scheduling at BS. The objectives of the algorithm are as follows: (i) To provide QoS to all classes of traffic (ii) To fairly allocate resources within each service class (iii) To ensure that lower priority flows could not affect higher priority flows (iv) To prevent starvation of lower priority flows (v) To ensure high resource utilization.

The performance of TLSA is extensively analyzed by simulations. The results show that the proposed algorithm can effectively and efficiently achieve the desired objectives. The remainder of the paper is organized as follows. Section II gives an overview of the related work. In section III we provide the details of TLSA. In section IV simulation results are provided to show the performance of the proposed solution. Section V concludes the paper.

II. EXISTING WORK

J. Chen, W. Jioa and H. Wang [2] provide a hierarchal scheduling scheme. They propose to use deficit fair priority queuing (DFPQ) to distribute bandwidth among service classes. Then earliest deadline first (EDF) algorithm is used for bandwidth allocation among rtPS flows, weighted fair queuing (WFQ) is used for nrtPS flows and round-robin (RR) for BE flows. However, the scheme cannot guarantee fair distribution of bandwidth among rtPS flows.

In [3] K. Wongthavarawat and A. Ganz propose a combination of strict priority allocation, EDF and WFQ. Uplink bandwidth is distributed according to strict priority: UGS, rtPS, nrtPS and BE. They propose EDF for rtPS class, while WFQ for nrtPS class. After distributing bandwidth to rtPS and nrtPS connections, the remaining bandwidth is distributed equally among active BE connections. A similar scheme is also proposed by V. Rangel, J. Ortiz and J. Gomez [4]. Similarly DN Lai, TC Huang and HY Chi [5] propose EDF for rtPS, WFQ for nrtPS, while first-come first-serve (FCFS) for BE. The co-scheduling is done according to strict priority. Lower priority flows can only get bandwidth if some bandwidth is not utilized by higher priority flows. These schemes do not guarantee fair distribution of bandwidth among flows of same service class. Furthermore, lower priority classes can starve due to strict priority allocation.

A. Sayenko, O. Alanen and J. Karhula [6] propose a scheme similar to weighted round robin (WRR). The scheme treats each connection as a separate session. The QoS requirements are used to determine the required number of frame slots, which then become the weights for WRR. The scheduling scheme comprises three stages: (i) Allocation of minimum number of slots (ii) Allocation of unused slots (iii) Ordering of slots to improve the provisioning of the QoS. The first stage is mandatory, while the other two are optimization steps. The calculation of number of slots for rtPS and nrtPS is almost identical, and the algorithm does not take into account the deadlines of rtPS packets.

A two-tier scheduling algorithm [7] is proposed by L. Chan, H. Chao and Z. Chou. The algorithm classifies connections into three categories:

1) Unsatisfied: a connection is unsatisfied if the allocated bandwidth is less than its minimum reserved traffic rate (MRTR).
2) Satisfied: a connection is classified as satisfied if its allocated bandwidth is between the MRTR and the maximum sustained traffic rate (MSTR).
3) Over-Satisfied: a connection is over-satisfied if the allocated bandwidth is greater than the MSTR.

The algorithm calculates weight for each connection based on its category and the QoS parameters. The bandwidth is first allocated to unsatisfied connections, then to satisfied connections, and the remaining bandwidth is then allocated to over-satisfied connections. No distinction is made on the service classes of the connections. Therefore, it may be difficult for the algorithm to ensure QoS for realtime applications.

Y. Shang and S. Cheng [8] provide a scheduling scheme that uses four scheduling servers: hard-QoS server, soft-QoS server, best-effort server and co-scheduling server. All servers implement WF^2Q scheduling. UGS traffic is scheduled by hard-QoS server. rtPS and nrtPS are scheduled by soft-QoS server, while BE service is scheduled by best-effort server. The co-scheduling server chooses one packet at a time from one of these servers to transmit in the uplink direction. No distinction is made between rtPS and nrtPS traffic and both are scheduled by the same server. Therefore under heavy loads, there is a risk that rtPS packets may miss their deadlines.

A token bucket based scheduling mechanism is presented in [9] by T.C. Tsai and C.Y. Wang. To avoid starvation of lower-priority classes, they set a threshold for each service class. When a service class gets more bandwidth than its threshold, then its priority is decreased. EDF is proposed for rtPS flows, while WRR is proposed for nrtPS flows. However, the fairness of bandwidth allocation is not considered by the authors.

R. Fei, K. Yang, S. Ou, S. Zhong and L. Gao [10] propose a dynamic bandwidth allocation algorithm. They provide a utility function that considers the QoS requirements of each service class. Each class is assigned a weight, which is then used by the utility function to determine the optimal scheduling. The algorithm is designed for relay mode operation and it may not be efficient for standard point-to-multipoint WiMAX networks.

X. Zhang, G. Zhang and H. Sun [11] provide a scheduling algorithm for fixed WiMAX. They propose to use WFQ. They claim that the algorithm can efficiently distribute bandwidth among rtPS flows, while indirectly guarantees the delay. The algorithm does not support QoS for nrtPS traffic. The algorithm is designed for fixed WiMAX and further research is required to make it applicable to mobile WiMAX.

C. Cicconetti, L. Lenzini, E. Mingozzi and C. Eklund [12] argue that BS uplink scheduling can be efficiently done by latency-rate schedulers. They propose the use of WRR for this purpose. However, the details of class specific scheduling are not provided.

III. METHODOLOGY

A. Call Admission Control

A new connection is admitted by the BS, if and only if the available bandwidth is sufficient to guarantee the MRTR for the incoming connection. To ensure that a connection never surpasses its contract, it is assumed that a traffic limiting module is present at the SS that always keeps the bandwidth demand below the MSTR. For a BE connection the MRTR is zero, and therefore it is always accepted.

B. Scheduling

To schedule traffics with different priorities and QoS requirements, we propose a two-level BS uplink scheduling scheme. At the first level, the available uplink bandwidth is distributed among different service classes. Then at the second level, class-specific algorithms are used to distribute the allocated bandwidth among the active connections of the same class.

1) First Level Scheduling (FLS): The objective of FLS is to distribute available bandwidth among different service classes, while ensuring following conditions:

1) QoS is ensured for all classes of traffic
2) Lower priority flows could not affect higher priority flows
3) Lower priority traffic is not starved
4) High bandwidth utilization

Priority is enforced by the order in which bandwidth is allocated. Bandwidth allocation is done in the following order: UGS, ertPS, rtPS, nrtPS, and BE. Thus UGS has the highest priority, while BE has the lowest priority.

Scheduling of UGS and ertPS traffic are similar and well-defined by the standard. Therefore FLS dynamically schedules rtPS, nrtPS, and BE traffics. The algorithm works as follows. Let F_o be the set of all active connections of service class o and let r_i^{min} be the MRTR for connection i. Then $\sum_{i \epsilon F_o} r_i^{min}$ amount of bandwidth is reserved for service class o. Henceforth, the reserved bandwidth for o will be represented by R_o. Since the MRTR is zero for a BE connection, therefore no bandwidth is reserved for a specific BE connection. Nevertheless, to prevent starvation of BE connections, a small percentage of total uplink bandwidth, R_{be}, is reserved for BE class.

Let r^{up} be the available uplink bandwidth after scheduling UGS and ertPS traffic. Then, at the start of each frame, $r^{up} - R_{nrtPS} - R_{be}$ bandwidth is available for rtPS connections. Note that this is the maximum amount of bandwidth that can be allocated to rtPS flows. However, if the total bandwidth request for rtPS class is less than the available bandwidth, then the remaining bandwidth can be allocated to nrtPS and BE flows. Thus the total bandwidth available to nrtPS connections is equal to R_{nrtPS} plus any unutilized bandwidth by rtPS class. After allocation of rtPS and nrtPS traffic, the remaining bandwidth can be allocated to BE connections. Obviously, at least R_{be} bandwidth is always available for BE class.

2) Second-Level Scheduling (SLS):

rtPS Class: For distributing bandwidth among rtPS flows, we propose to use the algorithm presented in [13]. The algorithm ensures fair and efficient allocation of bandwidth among rtPS connections. To ensure fairness the parameters of service ratios are defined as follows.

$$SR_i = \frac{\sum_{t=1}^{f-1} ba_i^t}{\sum_{t=1}^{f-1} br_i^t} \tag{1}$$

where, $i \epsilon F_{rtPS}$

$$SR = \frac{\sum_{t=1}^{f-1} \sum_{i=1}^{n} ba_i^t}{\sum_{t=1}^{f-1} \sum_{i=1}^{n} br_i^t} \tag{2}$$

where ba_i^t is the bandwidth allocated to connection i in frame t, while br_i^t is the bandwidth requested by i at the start of frame t, f is the current frame and n is the number of active rtPS connections. In each frame, to ensure fairness, bandwidth is allocated to i only if $SR_i \leq SR$.

Moreover, the algorithm keeps track of packet deadlines and makes sure that the packets do not miss their deadlines. The runtime complexity is $O(1)$. Further details of the algorithm can be found in [13].

nrtPS Class: The nrtPS allocation is done in two stages. Firstly, the algorithm makes sure that every connection gets at least its MRTR. In the second stage, the algorithm allocates more bandwidth to connections with greater backlog. Let for $u \epsilon F_{nrtPS}$, r_u^{cur} be the current bandwidth demand. Then, $\forall u$, the algorithm first allocates $min(r_u^{cur}, r_u^{min})$ amount of bandwidth to u. Let $bLog_u$ be the backlog of u after allocation in the first stage and r_{avl} be the amount of bandwidth still available in f for nrtPS flows. In the second stage, r_{avl} is distributed among nrtPS connections in proportion of their backlogs. Mathematically, the total bandwidth assigned to u is given as follows:

$$min(r_u^{cur}, r_u^{min}) + min(r_{avl}, \sum_{u \epsilon F_{nrtPS}} bLog_u) \times (\frac{bLog_u}{\sum_{u \epsilon F_{nrtPS}} bLog_u}) \tag{3}$$

The scheme ensures that each nrtPS connection gets at least the MRTR. Furthermore, using $bLog_u$ as weight enables the algorithm to accelerate data transmission for more demanding connections.

BE Class: The allocation of bandwidth at physical layer is done in terms of number of time slots. An SS with bad channel conditions consume more time slots for transmitting relatively small amount of data. We propose to distribute available time slots equally among BE connections so as to maximize the usage of bandwidth. Let C be the number of available time slots for BE traffic, and n_{be} be the number of BE connections. Then the number of slots available per connection can be given as C/n_{be}. For a BE connection v, let r_v^{cur} be the current bandwidth request and C_v time slots are required to fulfill the request. Then the algorithm allocates $min(C_v, C/n_{be})$ time slots to v. An SS with good channel conditions will be able to send more data within same number of time slots than an SS with poor channel conditions and thus automatically get prioritized. This scheme thus prevents SS with poor channel conditions to affect other SSs.

The difference between equal bandwidth allocation and equal time slot allocation can be illustrated with the help of an example. Suppose there are four SS: $S1$, $S2$, $S3$ and $S4$ with one BE connection each. Let the first three SSs have good channel conditions and in each time slot they can send 5 units of data , while $S4$ has poor channel conditions and it can send only 1 unit of data per slot. We also assume that 16 time slots are available for BE traffic. Then the bandwidth allocation under the two schemes is shown in Fig. 1. Under equal bandwidth distribution $S4$ is able to reduce the bandwidth of other connections by 50%. There is no QoS to guarantee and $S1$, $S2$ and $S3$ have good channel conditions but still they are paying the penalty of poor channel conditions of $S4$. Clearly, equal slot allocation makes use of bandwidth much more efficiently.

IV. Simulation Results

The performance of the proposed scheme is evaluated by simulations. Qualnet v5.02 [14] is used to perform all

(a)

	S1	S2	S3	S4	Total
Bandwidth Allocated	10	10	10	10	40
Time slots Required	2	2	2	10	16

(b)

	S1	S2	S3	S4	Total
Time slots Allocated	4	4	4	4	16
Bandwidth Available	20	20	20	4	64

Fig. 1. (a) Equal bandwidth distribution (b) Equal time slot distribution

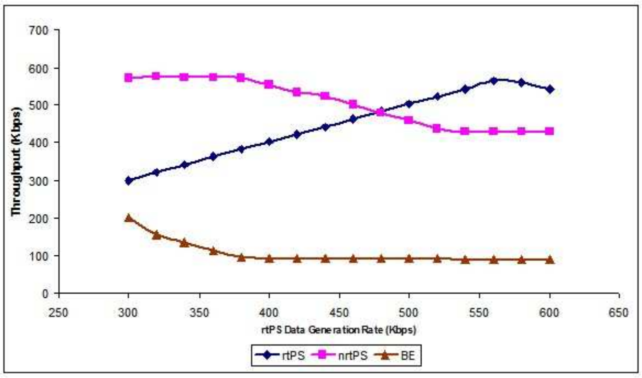

Fig. 2. Bandwidth distribution by FLS

Parameter	Value
Total uplink bandwidth	1 Mbps
Frame duration	20 ms
MAC propagation delay	1 μs
Cyclic prefix	8.0
Antenna model	omni antenna
Sampling factor	144/125
Propagation model	Two ray ground
Timeout interval	15 s
Antenna height	1.5 m
Antenna gain	1
Transmit power	20 dBm
Receive power threshold	205e-12
Carrier sense power threshold	0.9 * Receive power threshold
Link adaptation	Enabled

TABLE I
IMPORTANT SIMULATION PARAMETERS

simulations. Qualnet is a commercial network simulator that provides good support for the IEEE 802.16 standard. The important parameters used for simulation are presented in table I.

A. Experiment 1: Performance analysis of FLS

The purpose of this experiment is to assess the performance of FLS algorithm. For the simulation, BE traffic is generated at an average traffic rate of 200 Kbps throughout the experiment. Approximately 100 Kbps of bandwidth is reserved for BE traffic to prevent it from starvation. While for nrtPS the MRTR is 420 Kbps and the average traffic rate is 580 Kbps. Simulations are performed with increasing load of rtPS traffic. Initially, the average traffic rate of rtPS is 300 Kbps, which is gradually increased to 600 Kbps. The MRTR for rtPS traffic is 300 Kbps throughout the experiment, while the maximum allowed delay is set to 160ms.

The bandwidth distribution by FLS is shown in 2. As the rtPS traffic rate is increased from 300 Kbps to 400 Kbps, the throughput of BE traffic is reduced from 200 Kbps to 100 Kbps. While there is no effect on throughput of nrtPS traffic. As the rtPS data rate is further increased, the throughput of nrtPS decreases. Since 100 Kbps is the reserved bandwidth for BE class, therefore the throughput of BE traffic cannot be further reduced and remains unaffected. When rtPS throughput is 540 Kbps, the throughput of nrtPS reaches the MRTR i.e. 420 Kbps. Further increase in rtPS traffic rate have no effect on the throughput of nrtPS and BE traffic. So the throughput of rtPS cannot be further increased by just increasing its traffic generation rate.

It can be seen that FLS is able to ensure that rtPS and nrtPS classes get at least their MRTR. In case of overload rtPS gets the priority and FLS take away extra bandwidth from nrtPS and BE flows.

We also perform simulations to see if FLS is able to meet the deadlines of rtPS traffic. The end-to-end delay observed by different service classes is shown in Fig 3. It can be seen that rtPS traffic observed the least delay. In fact, the end-to-end delay of rtPS traffic remained around 30 ms throughout the experiment, while the maximum allowed delay is 160 ms. Increase in rtPS throughput result in lesser bandwidth allocation to nrtPS and BE services, which in turn result in higher delays for these services.

Next we analyze the performance of class specific algorithms. The results of performance evaluation of rtPS class algorithm are already presented in [13]. Therefore, here we only present the performance evaluation of SLS for nrtPS and BE classes. Nevertheless, we provide some results about rtPS class-specific algorithm to complement the results presented in [13].

B. Experiment 2: Class specific scheduling of nrtPS class

The experiment is performed to analyze the performance of nrtPS allocation algorithm under overload conditions. Four nrtPS connections with parameters as shown in table II are used in the simulation. Note that the only type of traffic present is nrtPS and the ratio of available bandwidth to the applied load is 0.88.

The corresponding bandwidth allocation is shown in Fig. 4. It can be seen that throughput remains at almost constant level for all connections. Furthermore, the MRTR is guaranteed for all nrtPS connections. We also calculated the service ratio (SR) as defined in equation 1. SR for N1, N2, N3 and N4 are approximately 0.89, 0.86, 0.87 and 0.87 respectively.

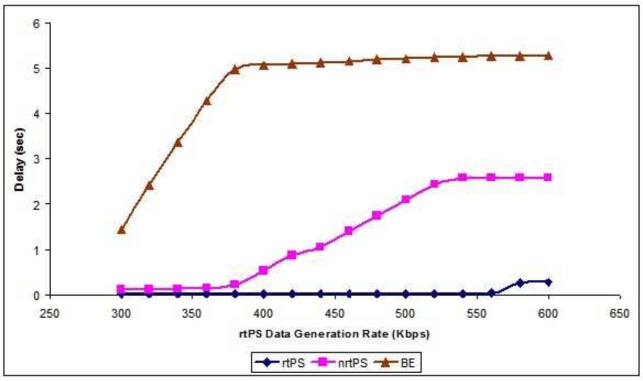

Fig. 3. End-to-end delay for different service classes under FLS

Connection	MRTR (Kbps)	Average Traffic Rate (Kbps)
N1	140	200
N2	210	300
N3	245	350
N4	225	320
Total	820	1170

TABLE II
INPUT TRAFFIC PARAMETERS FOR EXPERIMENT 2

This shows that the QoS is met for all connections and the bandwidth allocation is fair.

C. Experiment 3: Class specific scheduling of BE class

To analyze the bandwidth allocation among BE connections, four BE connections $BE1$, $BE2$, $BE3$ and $BE4$ are used. The average data generation rate is 200 Kbps, 300 Kbps, 350 Kbps, and 320 Kbps for $BE1$, $BE2$, $BE3$ and $BE4$ respectively. Again, the ratio of available bandwidth to applied load is 0.88 and only BE traffic is used for the experiment.

The throughput achieved by the connections is shown in Fig. 5. The algorithm equally divides the available time slots among active BE connections. However, the data generation rate of $BE1$ is smaller than the available bandwidth per connection. Therefore, the throughput of $BE1$ is equal to data generation rate. The remaining bandwidth is distributed among other three connections.

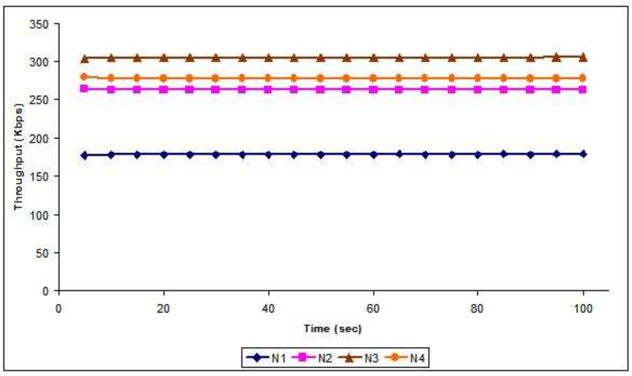

Fig. 4. Bandwidth allocation by nrtPS class specific algorithm

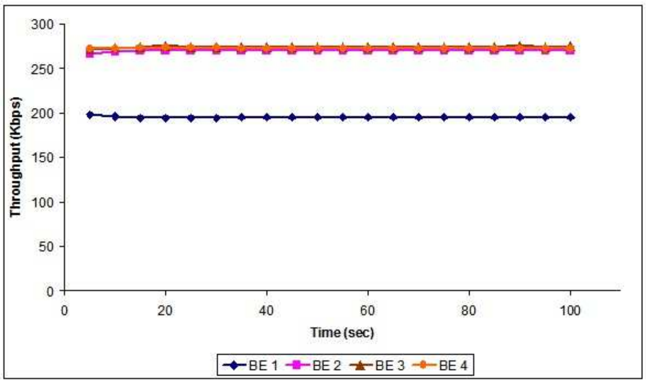

Fig. 5. Bandwidth allocation by BE class specific algorithm

We also noted the end-to-end delay for this experiment. We found that the delay was almost negligible for $BE1$, while it had the greatest value for $BE3$. At the end of simulation $BE1$, $BE2$, $BE3$ and $BE4$ have delay of 0.15s, 3.97s, 9.99s, and 6.6s respectively. Since for $BE1$, the throughput is equal to data generation rate, therefore the input queues remain almost empty and thus the waiting time in the queue is negligible. While, the difference of throughput and data generation rate is maximum for $BE3$. Therefore, more and more packets wait in the input queue with the passage of time and thus $BE3$ has large delay.

D. Experiment 4: Lost packets as function of load under mobility

The objective of this experiment is to determine the percentage of lost packets for rtPS class as function of load with mobile SS. The results of the experiment are presented in Fig. 6. In this simulation, the speed of SS is set to 60 Km/h (16.67 m/s) and it performs one handover. Simulations are performed with increasingly more load till the rtPS data generation rate is equal to total available uplink bandwidth. It can be seen that there is little increase in lost packets till the applied load is 80% of the available bandwidth. Further increase in load results in greater percentage of lost packets. However, the percentage always remain below 4%.

E. Experiment 5: Lost packets as function of SS speed for rtPS class

In this experiment an SS traverses a distance of 10 km and perform two handovers. The SS has an rtPS connection with average data generation rate of 200 Kbps. Fig. 7 shows the effect of SS speed on uplink transmission. It can be seen that there are no lost packets when the SS is stationary. The percentage of lost packets increases relatively quickly between the interval 0m/s to 10m/s. Further increase in SS speed, result in less significant increase in lost packets. The percentage of lost packets always remain below 1.2%. It should be noted that, in this simulation, the lost of some packets is due to

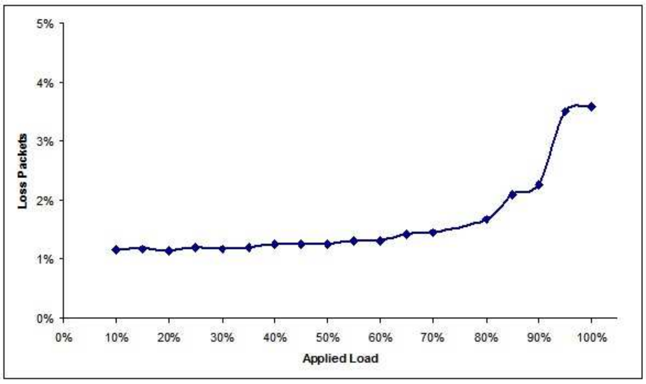

Fig. 6. Lost packets as function of traffic load under mobility

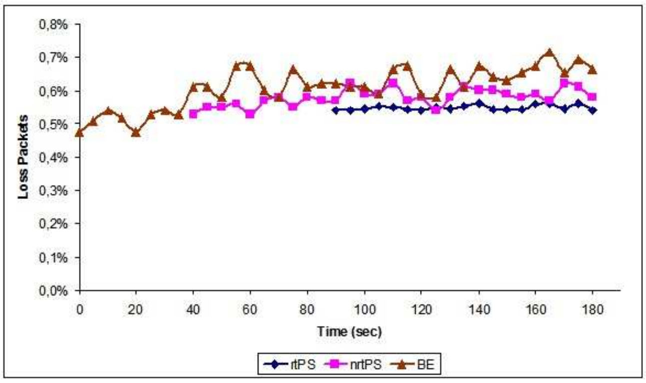

Fig. 9. The percentage of lost packets under mobility

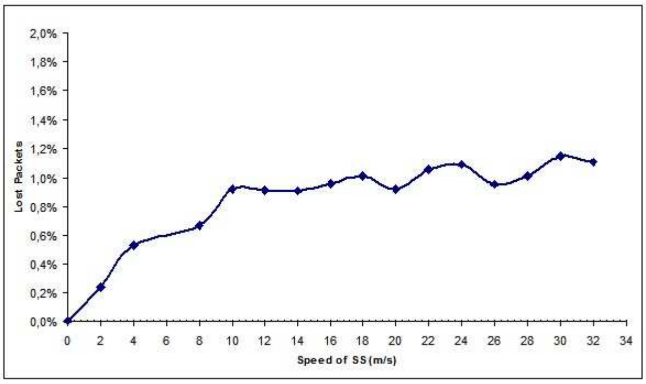

Fig. 7. Lost packets as function of SS speed

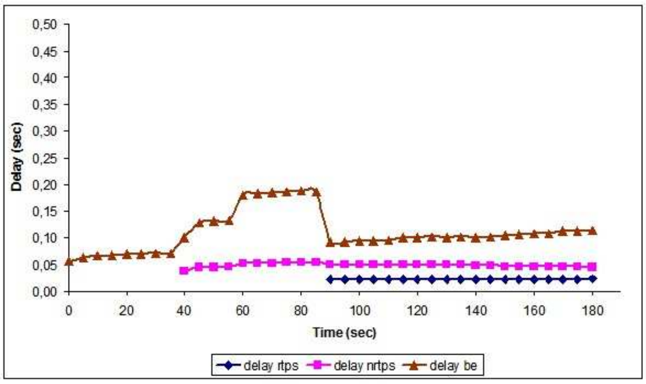

Fig. 10. Delay in mixed traffic network under mobility

physical layer phenomena and not because of the scheduling algorithm.

F. Experiment 6: Performance of FLS under mobility

The objective of this experiment is to asses the performance of FLS and observe the effect of mobility on different QoS parameters. Three SSs with one connection of each service class (rtPS, nrtPS and BE) are used. Each SS moves at a constant speed of 16.67 m/s and performs one handover during the simulation. Initially only BE traffic is present. The rate of BE traffic is gradually increased to 160 Kbps (0-40 sec). After 40th second the average rate of BE traffic is kept constant. At 40th second nrtPS traffic is introduced in the network. The rate of nrtPS traffic is gradually increase to 200 Kbps (40-85 sec). After this point, the average traffic rate of nrtPS is kept constant at 200 Kbps. rtPS traffic is introduced at this point and is increased gradually to 400 Kbps (85-180 sec).

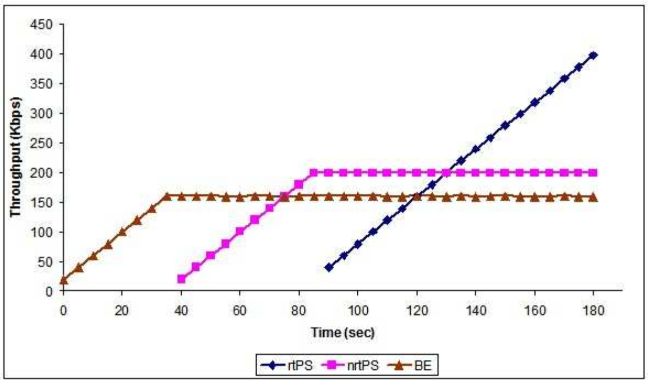

Fig. 8. Throughput of different classes of traffic

The throughput of all service classes at the receiving end is shown in Fig. 8. As the applied load is less than the available bandwidth, FLS is able to allocate bandwidth to service classes that exactly matches the input traffic pattern.

The percentage of lost packets is shown in Fig. 9. It can be seen that the percentage of lost packets remain below 0.75% for all classes of traffic. Furthermore, the fluctuation is the least in case of rtPS traffic. The percentage of lost packets is minimum for rtPS traffic, while maximum for BE traffic. However, the difference is not more than 0.1%. It can be seen that under normal load, the introduction of nrtPS traffic and rtPS traffic does not have significant effect on BE traffic.

The end-to-end delay for different classes of traffic is shown in Fig. 10. The introduction of nrtPS increases delay for BE

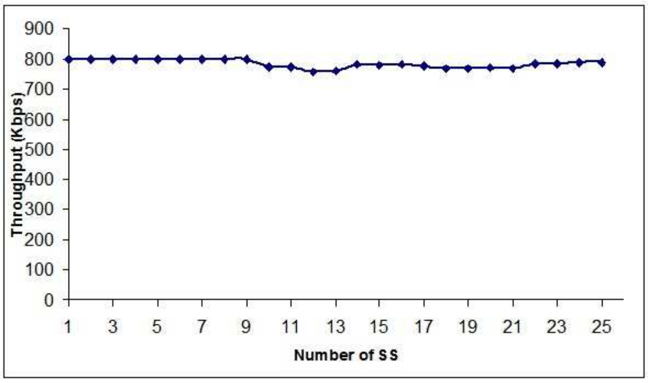

Fig. 11. Scalability of rtPS class specific algorithm

traffic. The delay of rtPS traffic remains constant irrespective of applied load and is around 25 ms, which is very good for realtime traffic. Also, the delay of nrtPS traffic remains below 60 ms throughout the experiment.

G. Experiment 7: Scalability

The objective is to determine the effect of number of SS on the performance of rtPS class specific algorithm. For this experiment, rtPS traffic is generated at an average rate of 800 Kbps. The experiment is performed for up to 25 SS. The average throughput achieved, shown in Fig. 11, suggests that the proposed solution is scalable.

V. Conclusion

In this paper, we proposed a two-level algorithm for the BS uplink scheduler. In the first level, bandwidth is distributed according to QoS requirements to different service classes. In the second level, class specific algorithms are used to distribute allocated bandwidth among flows of the same class. The simulation results show that the algorithm can guarantee QoS for all service classes, avoid starvation of BE traffic, and ensures fair allocation of bandwidth within same class.

References

[1] IEEE, "URL: http://standards.ieee.org/getieee802/802.16.html," last seen on 16 Feb 2012.

[2] J. Chen, W. Jiao, and H. Wang, "A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode," in *IEEE International Conference on Communications*, vol. 5, may 2005, pp. 3422 – 3426 Vol. 5.

[3] K. Wongthavarawat and A. Ganz, "Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems," *International Journal of Communication Systems*, no. 16, pp. 81–96, 2003.

[4] V. Rangel, J. Ortiz, and J. Gomez, "Performance analysis of QoS scheduling in broadband IEEE 802.16 based networks," in *OPNETWORK Technology Conference*, Washington D.C., 2006.

[5] D.-N. Lai, T.-C. Huang, and H.-Y. Chi, "Efficient bandwidth allocation with QoS guarantee for IEEE 802.16 systems," in *International Conference on Parallel Processing (ICPP)*, sept. 2011, pp. 115 –119.

[6] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2006, pp. 108–117. [Online]. Available: http://doi.acm.org/10.1145/1164717.1164737

[7] L. Chan, H. Chao, and Z. Chou, "Two-tier scheduling algorithm for uplink transmissions in IEEE 802.16 broadband wireless access systems," in *International Conference on Wireless Communications, Networking, and Mobile Computing*, 2006, pp. 1–4.

[8] Y. Shang and S. Cheng, "An enhanced packet scheduling algorithm for QoS support in IEEE 802.16 wireless networks," in *Third International Conference on Networking and Mobile Computing*, 2005, pp. 652–661.

[9] T. Tsai and C. Wang, "Routing and admission control in IEEE 802.16 distributed mesh networks," in *IFIP International Conference on Wireless and Optical Communications Networks*, Singapore, 2007, pp. 1–5.

[10] R. Fei, K. Yang, S. Ou, S. Zhong, and L. Gao, "A utility-based dynamic bandwidth allocation algorithm with QoS guarantee for IEEE 802.16j-enabled vehicular networks," in *Eighth International Conference on Embedded Computing SCALCOM-EMBEDDEDCOM'09*, sept. 2009, pp. 200 –205.

[11] X. Zhang, G. Zhang, and H. Sun, "A bandwidth allocation algorithm and its performance analysis based on IEEE 802.16d standard," in *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'11)*, sept. 2011, pp. 1 –4.

[12] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of service support in IEEE 802.16 networks," *IEEE Network*, no. 20, pp. 50–55, 2006.

[13] Z. Ahmed and S. Hamma, "Efficient and fair scheduling of rtPS traffic in IEEE 802.16 point-to-multipoint networks," in *4th Joint IFIP/IEEE wireless and mobile networking conferece*, Oct. 2011.

[14] Qualnet simulator (version 5.02). [Online]. Available: http://www.scalable-networks.com/products/qualnet/